Decoupling Abstractions of Non-linear Ordinary Differential Equations

Andrew T Sogokon, Khalil Ghorbal, Taylor T Johnson

▶ **To cite this version:**

Andrew T Sogokon, Khalil Ghorbal, Taylor T Johnson. Decoupling Abstractions of Non-linear Ordinary Differential Equations. FM 2016 - 21st International Symposium of Formal Methods, Nov 2016, Limassol, Cyprus. pp.628-644, 10.1007/978-3-319-48989-6_38 . hal-01374899v2

HAL Id: hal-01374899
https://hal.science/hal-01374899v2

Submitted on 28 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Decoupling Abstractions of Non-linear Ordinary Differential Equations *

Andrew Sogokon[1], Khalil Ghorbal[2], and Taylor T. Johnson[3]

[1] Vanderbilt University, Nashville, TN, USA
andrew.sogokon@vanderbilt.edu
[2] INRIA, Rennes, Brittany, France
khalil.ghorbal@inria.fr
[3] Vanderbilt University, Nashville, TN, USA
taylor.johnson@vanderbilt.edu

Abstract. We investigate decoupling abstractions, by which we seek to simulate (i.e. abstract) a given system of ordinary differential equations (ODEs) by another system that features completely independent (i.e. uncoupled) sub-systems, which can be considered as separate systems in their own right. Beyond a purely mathematical interest as a tool for the qualitative analysis of ODEs, decoupling can be applied to verification problems arising in the fields of control and hybrid systems. Existing verification technology often scales poorly with dimension. Thus, reducing a verification problem to a number of independent verification problems for systems of smaller dimension may enable one to prove properties that are otherwise seen as too difficult. We show an interesting correspondence between Darboux polynomials and decoupling simulating abstractions of systems of polynomial ODEs and give a constructive procedure for automatically computing the latter.

Keywords: ordinary differential equations, Darboux polynomials, simulation, abstraction, decoupling

1 Introduction

Simulation relations are an important concept in the study of both discrete and continuous dynamical systems. Informally speaking, a system simulates another system if it over-approximates its set of possible behaviours. In practice, when analyzing systems, one often wants to construct simulations of the original system that are in some sense "simpler" to analyze. Then, by demonstrating some property of interest in the simulation one may infer the property in the original system.

In [22] Sankaranarayanan investigated an interesting technique for constructing simulations of continuous systems by employing change of basis transformations. It was shown how *linearizing* change of basis transformations of non-linear

* This work was supported by the Air Force Research Laboratory (AFRL) through contract number FA8750-15-1-0105 and the Air Force Office of Scientific Research (AFOSR) under contract numbers FA9550-15-1-0258 and FA9550-16-1-0246.

systems of ODEs can yield simulations in which the dynamics is given by a system of linear ODEs. The motivation for considering such transformations is clear, since linear systems cannot exhibit some of the rich dynamic phenomena found in their non-linear counterparts and are more amenable to analysis [11]. In this paper we consider simulations of non-linear ODEs of a different kind: instead of linear dynamics, we seek to construct simulations that are potentially non-linear, but whose analysis can be performed in a lower-dimensional space than that of the original system.

Although our focus in this paper is on analyzing purely continuous systems, the methods we present are motivated by the broader goal of aiding the task of automatic verification of *hybrid dynamical systems* whose continuous modes are governed by non-linear ODEs. Hybrid systems combine discrete and continuous behaviour; their formal modelling and verification is of increasing interest and importance to modern engineering, where discrete digital controllers are used to control continuously evolving physical plants. In recent years, verification technology for hybrid systems has seen significant advances and a number of interesting case studies have been reported, e.g. verification of train control systems [20,29], aircraft collision avoidance protocols [13,1], descent guidance control software in a lunar lander [28] and satellite rendezvous manoeuvres [14], to give a few examples. However, non-linear ODEs appearing in hybrid system models often present a serious challenge to verification due to their inherent complexity. In this paper we seek to overcome some aspects of this hurdle by constructing simulations of non-linear ODEs with structure that more readily lends itself to analysis.

1.1 Contributions

In this paper we `(I)` define *decoupled simulating abstractions* of non-linear ODEs, discuss their utility and relationship to *first integrals* [10] and *constant-scale continuous consecutions* [23]. `(II)` We give an algorithm for checking whether a given set of polynomial *abstract basis functions* can be used to create a decoupled abstraction of a system of polynomial ODEs and then `(III)` employ the theory of *Darboux polynomials* [10] to give sufficient criteria for non-existence of polynomial abstract basis functions suitable for constructing decoupled polynomial abstractions. Lastly, `(IV)` we show how Darboux polynomials can be used to construct the abstract basis functions for decoupled abstractions whenever they exist. We conclude with a summary of our findings, an overview of related work and directions for future research.

1.2 Preliminaries

An autonomous n-dimensional system of ODEs has the following form:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, \ldots, x_n), \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \ldots, x_n), \end{aligned}$$

where for $i \in \{1, \ldots, n\}$ each $f_i : \mathbb{R}^n \to \mathbb{R}$ is a real-valued function (typically C^1), and $\dot{x}_i$ denotes the time derivative of x_i, i.e. $\frac{d}{dt}x_i(t)$. In applications, constraints are often imposed on the states where the system is allowed to evolve, i.e. the system may only evolve inside some given set $H \subseteq \mathbb{R}^n$, which is known as the *evolution constraint*. We may write this more concisely using vector notation as $\dot{\boldsymbol{x}} = f(\boldsymbol{x}),\ \boldsymbol{x} \in H$. Here $\dot{\boldsymbol{x}} = (\dot{x}_1, \ldots, \dot{x}_n)$ and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a *vector field* generated by the system, i.e. $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \mathbb{R}^n$. When no evolution constraint is specified, H is assumed to be $\mathbb{R}^n$.

A *solution* to the initial value problem for the system of ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ with initial value $\boldsymbol{x}_0 \in \mathbb{R}^n$ is a differentiable function $\varphi_t(\boldsymbol{x}_0) : (a, b) \to \mathbb{R}^n$ defined for all t within some non-empty extended real interval including zero, i.e. $t \in (a, b) \subseteq \mathbb{R} \cup \{\infty, -\infty\}$, where $a < 0 < b$, and such that $\frac{d}{dt}\varphi_t(\boldsymbol{x}_0) = f(\varphi_t(\boldsymbol{x}_0))$ for all $t \in (a, b)$. If the solution $\varphi_t(\boldsymbol{x}_0)$ is available in closed-form,[4] then one can answer questions about the temporal behaviour of the system (such as e.g. safety and liveness) by analyzing the closed-form expression. In practice, however, it has long been established that explicit closed-form solutions to non-linear ODEs are highly uncommon [11].

In this paper we will work with systems of ODEs whose right-hand sides are given by polynomials in the state variables $x_1, \ldots, x_n$. Formally, we say that $f_i \in \mathbb{R}[X_1, \ldots, X_n]$ for all $i \in \{1, \ldots, n\}$, where $\mathbb{R}[X_1, \ldots, X_n]$ denotes the ring of multivariate polynomials with real coefficients and indeterminates $X_1, \ldots, X_n$. We write $f_i(x_1, \ldots, x_n)$ when we wish to make it clear that the polynomial is treated as a function, with indeterminates replaced by the appropriate variables. Polynomial systems of ODEs are necessarily locally Lipschitz continuous, which guarantees existence of unique solutions on some non-trivial time interval for any initial value $\boldsymbol{x}_0 \in \mathbb{R}^n$ (by the Picard-Lindelöf theorem; see e.g. [27]).

1.3 Coupling

Given a system of ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, the *maximum coupling coefficient* (henceforth mcc) is the size of the largest sub-system with no independent sub-systems. To define rigorously, we construct a finite *coupling graph* $CG = (V, E)$, where the set of vertices is precisely the set of state variables, i.e. $V = \{x_1, \ldots, x_n\}$, and there is an edge from x_i to some other vertex x_j, i.e. $(x_i, x_j) \in E$ with $i \neq j$, if and only if $\frac{\partial f_i}{\partial x_j} \neq 0$. The *coupling coefficients* cc are a finite multiset of natural numbers corresponding to the orders (i.e. the numbers of vertices) of all the weakly connected components in CG. The coefficient mcc is defined to be the maximum order of the weakly connected components in CG, i.e. $\mathsf{mcc} \equiv \max \mathsf{cc}$.

Definition 1 (Uncoupled system). *A system of ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ is* uncoupled *if and only if its $\mathsf{mcc} = 1$, i.e. if the rate of change of each state variable is completely independent of the other variables.*

[4] By this we understand a *finite* expression in terms of polynomials and elementary functions such as $\sin, \cos, \exp, \ln$, etc.

Example 1. Consider the following two planar polynomial systems:

$$\begin{aligned} \dot{x}_1 &= x_1^2 x_2 + 5x_1 - 1, & \dot{x}_1 &= x_1^3 + 5x_1 - 10, \\ \dot{x}_2 &= 3x_2^3 + 2x_1 x_2 - x_1. & \dot{x}_2 &= 2x_2^2 + 3x_2 + 1. \end{aligned}$$

The system on the left has $\mathsf{mcc} = 2$ because the vertices $\{x_1, x_2\}$ in the coupling graph have edges connecting them in both directions, since $\frac{\partial}{\partial x_2}(x_1^2 x_2 + 5x_1 - 1) = x_1^2 \neq 0$ and $\frac{\partial}{\partial x_1}(3x_2^3 + 2x_1 x_2 - x_1) = 2x_2 - 1 \neq 0$. On the other hand, the system on the right has $\mathsf{mcc} = 1$ (i.e. is uncoupled) because $\frac{\partial}{\partial x_2}(x_1^3 + 5x_1 - 10) = 0$ and $\frac{\partial}{\partial x_1}(2x_2^2 + 3x_2 + 1) = 0$ and therefore the vertices $\{x_1, x_2\}$ in the graph are disconnected.

Uncoupled systems are appealing first and foremost because their 1-dimensional sub-systems can be analyzed independently, following a standard technique for 1-dimensional flows (see e.g. [25, Chapter 2]). For instance, consider the 1-dimensional system $\dot{x} = x^3 + 5x^2 + x - 10$. This system evolves on the real line and has fixed points at the real roots of $x^3 + 5x^2 + x - 10$, of which there are three: $\{-2, \frac{1}{2}\left(-3 - \sqrt{29}\right), \frac{1}{2}\left(-3 + \sqrt{29}\right)\}$. The direction of the flow is to the right whenever the graph of $\dot{x}$ is above zero (i.e. the rate of change of x is positive) and to the left when it is below (the rate of change is negative), as shown in Figure 1.3.

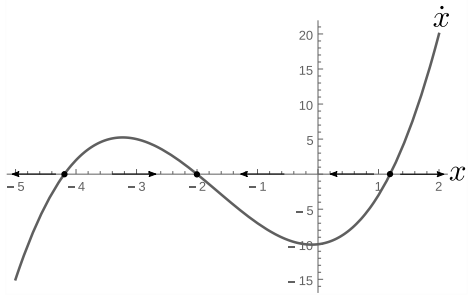

Fig. 1. Analysis of the 1-dimensional system $\dot{x} = x^3 + 5x^2 + x - 10$.

From inspecting the figure, one can readily see how one can construct the set of reachable states of any given initial point x_0 in a 1-dimensional polynomial system $\dot{x} = f(x)$: either the point is a root of the right-hand side, i.e. $f(x_0) = 0$, in which case x_0 remains invariant and the reachable set is simply $\{x_0\}$, or x_0 is not a root, i.e. $f(x) \neq 0$, in which case the reachable set is an interval of the form $[x_0, r)$ or $(r, x_0]$, where $r \in \mathbb{R} \cup \{\infty, -\infty\}$ is either a real root of f or it is ∞ or $-\infty$, respectively (if there are no real roots in the direction of motion). The reachable set from any initial point $\boldsymbol{x}_0 \in \mathbb{R}^n$ in a uncoupled system can thus also be bounded by combining the independent reachable sets in the 1-dimensional sub-systems.

Bounded-time reachable set computation using *verified integration* methods is also made easier because large systems of non-linear ODEs are typically expensive to integrate using methods that yield tight enclosures [16] (such as *Taylor models* [3,17]), whereas in an uncoupled system, no matter how large, each 1-dimensional sub-system can be integrated separately. An enclosure of the solution to the whole system at some time t can then be constructed directly from the enclosures of the solutions to the sub-systems at that time.

2 Decoupled Simulating Abstractions

In what follows, we will adopt the approach described by Sankaranarayanan in [22] to define *simulating abstractions* of non-linear ODEs using appropriate change of basis transformations.

Definition 2 (Simulating abstraction). *For a system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}),\ \boldsymbol{x} \in H$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz continuous, equipped with an initial set of states $X_0 \subseteq \mathbb{R}^n$, a system $\dot{\boldsymbol{\alpha}} = G(\boldsymbol{\alpha}),\ \boldsymbol{\alpha} \in \widehat{H}$, where $G : \mathbb{R}^m \to \mathbb{R}^m$ is locally Lipschitz continuous and equipped with an initial set of states $\widehat{X}_0 \subseteq \mathbb{R}^m$ is a* simulating abstraction *if there exists a smooth (i.e. C^∞) mapping $\boldsymbol{\alpha} : \mathbb{R}^n \to \mathbb{R}^m$ such that: (i) $\boldsymbol{\alpha}(X_0) \subseteq \widehat{X}_0$, (ii) $\boldsymbol{\alpha}(H) \subseteq \widehat{H}$, and (iii) for any trajectory (i.e. solution in non-negative time) $\varphi_\tau(\boldsymbol{x}_0) : [0, T) \to H$ of the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}),\ \boldsymbol{x} \in H$, the trajectory $\boldsymbol{\alpha} \circ \varphi_\tau(\boldsymbol{x}_0) : [0, T) \to \widehat{H}$ is a trajectory of $\dot{\boldsymbol{\alpha}} = G(\boldsymbol{\alpha}),\ \boldsymbol{\alpha} \in \widehat{H}$.*

To ensure that the last condition in the above definition holds, it is sufficient to show that $G(\boldsymbol{\alpha}(\boldsymbol{x})) = J_{\boldsymbol{\alpha}} \cdot f(\boldsymbol{x})$, where $J_{\boldsymbol{\alpha}}$ is the Jacobian of the smooth mapping $\boldsymbol{\alpha}$ w.r.t. the state variables $x_1, \ldots, x_n$ (see [22, Theorem 2.1]), i.e.

$$G(\boldsymbol{\alpha}) = \begin{pmatrix} \frac{\partial \alpha_1}{\partial x_1} & \cdots & \frac{\partial \alpha_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \alpha_m}{\partial x_1} & \cdots & \frac{\partial \alpha_m}{\partial x_n} \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}.$$

Definition 3 (Lie derivative). *For a given system of ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, the* Lie derivative *of a smooth function $p : \mathbb{R}^n \to \mathbb{R}$ is given by*

$$\mathfrak{L}_f(p) = \nabla p \cdot f = \sum_{i=1}^{n} \frac{\partial p}{\partial x_i} \cdot f_i.$$

Note that since $f_i(\boldsymbol{x}) = \frac{dx_i}{dt}$, $\mathfrak{L}_f(p) = \left(\sum_{i=1}^{n} \frac{\partial p}{\partial x_i} \cdot \frac{dx_i}{dt}\right) = \frac{dp}{dt}$ i.e. the total derivative of the function p with respect to time, which we denote by $\dot{p}$.

Let us recall that the gradient ∇p gives the vector of all the partial derivatives of p, i.e. $\nabla p \equiv \left(\frac{\partial p}{\partial x_1}, \frac{\partial p}{\partial x_2}, \ldots, \frac{\partial p}{\partial x_n}\right)$, and thus the necessary condition for (iii) in Definition 2 to be satisfied may be equivalently stated as:

$$G(\boldsymbol{\alpha}) = \begin{pmatrix} \nabla \alpha_1 \\ \vdots \\ \nabla \alpha_m \end{pmatrix} \cdot f = \begin{pmatrix} \mathfrak{L}_f(\alpha_1) \\ \vdots \\ \mathfrak{L}_f(\alpha_m) \end{pmatrix}.$$

Remark 1. It is important to note that, following Def. 2, solutions to simulating abstractions are guaranteed to exist for at least as long as they do in the concrete system. This property is crucial to soundness of the abstraction. A rather different, but in a certain sense more general, concept was explored by Platzer, who introduced *differential ghosts* [19], where the original dynamics is augmented by introducing some fresh variables whose rate of change may feature the newly introduced variables themselves, but is not restricted in the same way as in Def. 2. However, extra care needs to be taken to ensure that the solutions of the newly defined dynamics exist for at least as long as the solutions to the original system (e.g. see [19, Proof of Theorem 38]).

Definition 4 (Decoupling simulating abstraction). *Given a system of ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, a simulating abstraction $\dot{\boldsymbol{\alpha}} = G(\boldsymbol{\alpha})$ is* decoupling *if and only if the equalities $\mathfrak{L}_f(\alpha_1) = G_1(\alpha_1), \ldots, \mathfrak{L}_f(\alpha_m) = G_m(\alpha_m)$ hold, where $(G_1, \ldots, G_m) = G$. Such an abstraction is thus uncoupled:*

$$\begin{aligned} \dot{\alpha}_1 &= G_1(\alpha_1), \\ &\vdots \\ \dot{\alpha}_m &= G_m(\alpha_m). \end{aligned}$$

In what follows, we will give some examples of how first integrals (see e.g. [10]) and constant-scale continuous consecutions [23] provide the abstract basis functions $\boldsymbol{\alpha}$ which lead to decoupling simulating abstractions.

Example 2 (Algebraically integrable system). The 3-dimensional system

$$\begin{aligned} \dot{x}_1 &= x_1(x_3 - x_2), \\ \dot{x}_2 &= x_2(x_1 - x_3), \\ \dot{x}_3 &= x_3(x_2 - x_1), \end{aligned}$$

has two independent polynomial conserved quantities, i.e. first integrals, given by $\alpha_1 = x_1x_2x_3$ and $\alpha_2 = x_1 + x_2 + x_3$ (see [8, Ex. 75]). If we let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, we obtain the decoupling simulating abstraction $\dot{\boldsymbol{\alpha}} = \mathbf{0}$, i.e. $\dot{\alpha}_1 = 0, \dot{\alpha}_2 = 0$.

Remark 2. A polynomial system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ of size n is *algebraically integrable* if it possesses $n-1$ independent polynomial conserved quantities (also known as first integrals; see [10,8]), i.e. polynomials $\{\alpha_1, \ldots, \alpha_{n-1}\}$, where for all $i = 1, \ldots, n-1$ one has $\mathfrak{L}_f(\alpha_i) \equiv \dot{\alpha}_i = 0$. Algebraic integrability is a very powerful property, since it allows one to construct tight approximations of the *orbit* $\gamma(\boldsymbol{x}_0)$, i.e. the reachable set from $\boldsymbol{x}_0 \in \mathbb{R}^n$ in positive as well as negative time. That is, for any given point $\boldsymbol{x}_0 \in \mathbb{R}^n$, if one evaluates each first integral $\alpha_1, \ldots, \alpha_{n-1}$ at $\boldsymbol{x}_0$, one obtains real constants $c_1, \ldots, c_{n-1}$. The orbit through $\boldsymbol{x}_0$ is guaranteed to satisfy the formula $\alpha_1 = c_1 \wedge \cdots \wedge \alpha_{n-1} = c_{n-1}$, which corresponds to a (real) algebraic subset of $\mathbb{R}^n$ given by the common real roots of the polynomials $\alpha_i - c_i$. Every point $\boldsymbol{\alpha}_0 \in \mathbb{R}^{n-1}$ in such an abstract system $\dot{\boldsymbol{\alpha}} = \mathbf{0}$ is invariant and corresponds to a real (and invariant) algebraic set containing the orbit of the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$.

Polynomials p such that $\mathfrak{L}_f(p) = \lambda p$ for some $\lambda \in \mathbb{R}$ generalize polynomial first integrals[5] and were investigated by Sankaranarayanan et al. in [23], where they were used in *constant-scale continuous consecution* conditions. In general, if one can find polynomials $\alpha_1, \ldots, \alpha_m$ that satisfy $\mathfrak{L}_f(\alpha_i) = \lambda_i \alpha_i$, $\lambda_i \in \mathbb{R}$ for all $i \in \{1, \ldots, m\}$, then one obtains a decoupling abstraction of the form

$$\begin{aligned} \dot{\alpha}_1 &= \lambda_1 \alpha_1, \\ &\vdots \\ \dot{\alpha}_m &= \lambda_m \alpha_m. \end{aligned}$$

We generalize this idea to decoupling polynomial abstractions by considering polynomial functions $\alpha_i \in \mathbb{R}[X_1, \ldots, X_n]$ such that $\mathfrak{L}_f(\alpha_i) = G_i(\alpha)$, where $G_i \in \mathbb{R}[X]$, i.e. the derivative of α_i may be expressed as a polynomial in α_i with real coefficients.

Example 3 (Decoupling simulating abstraction). Consider the coupled system:

$$\begin{aligned} \dot{x}_1 &= \frac{1}{3}(1 - 3x_1 + 2x_1^2 - 6x_2 + 4x_1x_2 + 2x_2^2), \\ \dot{x}_2 &= \frac{1}{3}(-1 - 3x_1 + x_1^2 + 2x_1x_2 + x_2^2)\,. \end{aligned}$$

Let $\alpha_1 = x_1 + x_2 - 1$, $\alpha_2 = x_1 - 2x_2$. If we consider $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, we arrive at the following system (left), which can be expressed as an uncoupled system in the new basis (right):

$$\begin{aligned} \dot{\alpha}_1 &= -2x_1 + x_1^2 - 2x_2 + 2x_1x_2 + x_2^2, & \dot{\alpha}_1 &= \alpha_1^2 - 1, \\ \dot{\alpha}_2 &= 1 + x_1 - 2x_2, & \dot{\alpha}_2 &= \alpha_2 + 1\,. \end{aligned}$$

3 Existence and Generation of Abstraction Polynomials

In what follows, we investigate the existence of polynomials that can be used to construct decoupling simulating abstractions of a given system. We show in Section 3.1 that their existence (to a given polynomial degree) is decidable and give a sufficient criterion for their non-existence (to a given degree) based on the existence of so-called *Darboux polynomials* (e.g. see [10]). We then explore the problems of *checking* and *generation*. The checking problem is concerned with determining whether a given candidate polynomial is suitable for constructing a decoupling simulating abstraction. In Section 3.2 we describe a procedure for solving the checking problem. In Section 3.3 we present a technique for generating all suitable polynomials for the decoupling abstract basis (up to a given polynomial degree).

[5] i.e. p is a first integral if $\mathfrak{L}_f(p) = \lambda p$ where $\lambda = 0$.

3.1 Decidability and Darboux Existence Criterion

For polynomial systems of ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, the problem of finding a non-constant polynomial in the state variables, $p \in \mathbb{R}[X_1, \ldots, X_n]$, for the decoupling abstract basis reduces to searching for those p such that $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$, i.e. G is a univariate polynomial with real coefficients. There may, however, be no such polynomial. Fortunately, it is decidable to check for existence of such a p.

Proposition 1 (Existence of decoupling abstract basis polynomials). *Given a positive integer d and a polynomial system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, it is decidable to check whether there exists a polynomial $p \in \mathbb{R}[X_1, \ldots, X_n]$ of total degree d such that $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$ is a univariate polynomial with real coefficients.*

Proof. The problem can be stated as a sentence in the theory of real arithmetic which is decidable [26]. Let $\lambda_0, \ldots, \lambda_k$ denote the unknown coefficients of the generic polynomial template p of degree d, where $k := \binom{n+d}{d} - 1$ is the number of *non-constant* monomials of degree at most d in n variables. The Lie derivative $\mathfrak{L}_f(p)$ can therefore be symbolically computed (Def. 3). Let $\kappa_0, \ldots, \kappa_m$ denote the unknown coefficients of the polynomial $G \in \mathbb{R}[X]$ where $m := \lceil \deg(\mathfrak{L}_f(p))/d \rceil$. The decision problem stated in the proposition is therefore equivalent to deciding the truth of the following sentence:

$$\begin{aligned} &\exists\, (\lambda_0, \ldots, \lambda_k) \in \mathbb{R}^{k+1}.\ \exists\, (\kappa_0, \ldots, \kappa_m) \in \mathbb{R}^{m+1}. \\ &\qquad \forall (X_1, \ldots, X_n) \in \mathbb{R}^n.\ d > 0 \wedge \mathfrak{L}_f(p) - (\kappa_0 + \kappa_1 p + \cdots + \kappa_m p^m) = 0 \ . \end{aligned}$$

If λ_0 denotes the constant term of the generic polynomial template p, then the condition $d > 0$ is equivalent (over the reals) to the inequality $\sum_{0<i\leq k} \lambda_i^2 > 0$, ensuring that p is non-constant. □

In practice, there is currently no question of applying existing decision procedures to formulas constructed in the proof or Prop. 1. The complexity of the most popular procedure for real quantifier elimination (CAD, due to Collins [4]) is doubly exponential in the number of variables. In Section 3.3 we will pursue a more promising method of searching for decoupling abstract basis polynomials. First, we shall recall so-called *Darboux polynomials*, a well-known tool in the study integrability of dynamical systems (e.g. see [10]), and use them to give a non-existence criterion for decoupling abstract basis polynomials. We then explore an interesting relationship between the two concepts.

Definition 5 (Darboux polynomial). *A polynomial $q \in K[X_1, \ldots, X_n]$, where K is a field of characteristic zero (e.g. $\mathbb{C}, \mathbb{R}, \mathbb{Q}$), is a* Darboux polynomial[6] *for $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ iff $\mathfrak{L}_f(q) = \lambda q$, for some $\lambda \in K[X_1, \ldots, X_n]$.*

[6] When q is a constant, the Darboux polynomial is *trivial*[10, Definition 2.14]. In this paper we will generally be interested in the non-trivial case.

Proposition 2 (Criterion for non-existence of decoupled abstractions). *If a given system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ does not admit any Darboux polynomials over $\mathbb{C}$ of degree d, then there is no polynomial $p \in \mathbb{R}[X_1, \ldots, X_n]$ of degree d such that $\mathfrak{L}_f(p) = G(p)$ for some* non-constant *$G \in \mathbb{R}[X]$.*

Proof. We prove the contrapositive. Suppose there exists a polynomial $p \in \mathbb{R}[X_1, \ldots, X_n]$ such that $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$ is non-constant. By the fundamental theorem of algebra, G must have at least one complex root $c \in \mathbb{C}$. Therefore $G = (X - c)H$, where $H \in \mathbb{C}[X]$. We see that $(p - c)$ is a Darboux polynomial for the system because

$$\mathfrak{L}_f(p - c) = \mathfrak{L}_f(p) - \mathfrak{L}_f(c) = \mathfrak{L}_f(p) = G(p) = (p - c)H(p).$$

The degree of the Darboux polynomial $p - c$ is equal to the degree of p. □

3.2 Checking Abstraction Polynomial Candidates

Before proceeding to methods for generating decoupling abstract basis polynomials for polynomial systems $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, we discuss the (easier) problem of checking if for a given $p \in \mathbb{R}[X_1, \ldots, X_n]$ one can write $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$.

In general, given any two polynomials $P, p \in \mathbb{R}[X_1, \ldots, X_n]$, if $\deg(P) \geq \deg(p)$, one may obtain a rewriting $P = G(p)$ by solving a system of linear equations. One proceeds by first defining the maximum degree of a possible G to be $d = \lceil \deg(P)/\deg(p) \rceil$. If an appropriate rewriting exists, then there is guaranteed to be a solution $(\lambda_0, \ldots, \lambda_d) \in \mathbb{R}^{d+1}$ to the equation $P = \lambda_0 + \lambda_1 p + \lambda_2 p^2 + \cdots + \lambda_d p^d$. By expanding and equating the monomial coefficients on both sides one arrives at a system of linear equations (of size no larger than the number of monomials of P) in the real variables $\lambda_0, \ldots, \lambda_d$. Thus, in the worst case, one has to solve a linear system with $d+1$ variables and $\binom{n+\deg(P)}{\deg(P)}$ equational constraints. A solution may be computed using a linear solver and the rewriting polynomial constructed as $G = \lambda_0 + \lambda_1 X + \cdots + \lambda_d X^d$. In what follows, we will refer to the procedure for obtaining the rewriting as Rewrite, that is Rewrite(P, p) gives G whenever $P = G(p)$.

Remark 3. It is worth remarking that the procedure Rewrite can be implemented by performing successive *polynomial reductions*, rather than by solving a linear program. Polynomial reduction extends polynomial division for univariate polynomials to the multivariate case and in general requires the computation of *Gröbner bases*. This functionality is available in most modern computer algebra systems.

3.3 Automated Generation of Decoupling Abstractions

A highly efficient method for synthesizing polynomial first integrals for polynomial ODEs was reported by Matringe et al. in [15], where the synthesis problem is reduced to computing the null space of a matrix with real entries. In [7], the

authors extended the work of Matringe et al. to generate real algebraic invariants of polynomial ODEs, giving a search procedure for the most general class of invariant sets that can be expressed using polynomial equations. The same procedure can be used to generate Darboux polynomials over the reals or over the complexes only by changing the underlying computational field. In general, there is no known bound for the degree of Darboux polynomials in a given system. However, the automatic generation procedure is guaranteed to find all the independent Darboux polynomials for the system up to a given degree.

In this section, we explore the relationship between polynomials in a decoupling abstract basis and Darboux polynomials. This relationship will enable us to exploit the efficient symbolic generation methods reported in [15,7]. We outline a procedure for constructing polynomials p such that $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$, from a list of automatically generated Darboux polynomials (up to some given degree). The procedure will require two lemmas given below.

We note first that whenever q is a Darboux polynomial, any constant multiple of q, i.e. aq for some $a \in \mathbb{R}$ or $\mathbb{C}$, is also Darboux. A similar property holds for the decoupling abstract basis functions in simulating abstractions.

Lemma 1. *If $p \in \mathbb{R}[X_1, \ldots, X_n]$ is such that $\mathfrak{L}_f(p) = G(p)$ where $G \in \mathbb{R}[X]$, then $s = ap + b$ for any real numbers a, b, is such that $\mathfrak{L}_f(s) = F(s)$, where $F \in \mathbb{R}[X]$.*

Proof. If $a = 0$ then $\mathfrak{L}_f(s) = \mathfrak{L}_f(b) = 0$ and F is simply the zero polynomial in $\mathbb{R}[X]$. If $a \neq 0$, by our hypothesis we have $\mathfrak{L}_f(p) = G(p)$. Let us write $p = \frac{s-b}{a}$ and note that

$$\mathfrak{L}_f(s) = \mathfrak{L}_f(ap + b) = a\mathfrak{L}_f(p) + \mathfrak{L}_f(b) = a\mathfrak{L}_f(p) = aG(p) = aG\left(\frac{s-b}{a}\right).$$

We see that $\mathfrak{L}_f(s) = aG\left(\frac{s-b}{a}\right)$ is a polynomial in s with real coefficients. □

One consequence of Lem. 1 is that whenever we assume the existence of a polynomial p such that $\mathfrak{L}_f(p) = G(p)$ for some $G \in \mathbb{R}[X]$, it always suffices to assume the existence of a decoupling abstract basis polynomial $p - r$ for any real number r.

In Prop. 2 we established that the existence of decoupling abstract basis polynomials p is related to the existence of a special Darboux polynomial $p - c$ for some complex number c. For any polynomial s, we denote by s^* the polynomial obtained by setting the constant term of s to zero. For instance, if $s = x + 1$ then, $s^* = x$. Thus, for the (Darboux) polynomial $p - c$, one has $(p - c)^* = p^*$ (by definition of the * operator) and therefore p^* is a decoupling abstract basis polynomial by Lem. 1, since it is an offset of the polynomial p by a real number (the constant term of p). Therefore, if one generates Darboux polynomials over the complex numbers and finds a Darboux polynomial q such that q^* is a polynomial over the reals (i.e. all the coefficients of q^* are real numbers), then q^* is potentially a decoupling abstract basis polynomial, which can be checked by solving a linear program, i.e. by running $\textsc{Rewrite}(\mathfrak{L}_f(q^*), q^*)$, as outlined in Section 3.2.

Nevertheless, generating Darboux polynomials over the complex numbers will not necessarily return Darboux polynomials q such that q^* is a polynomial over the reals even if the latter exist. For instance, if $q = x^2 + xy + c$ is a Darboux polynomial with some complex constant term c, then the procedure may return $\imath q$ instead of q ($\imath$ being the imaginary number satisfying $\imath^2 = -1$), although we are rather interested in looking for q. Enforcing such a constraint in the procedure for generating Darboux polynomials will require solving mixed non-linear equations where some variables are real and some are complex numbers. To avoid solving mixed problems, we can easily adapt the generation procedure to produce *monic* Darboux polynomials for any variable ordering, for instance the lexicographic order $X_1 > \cdots > X_n$. Recall that monic univariate polynomials are those polynomials where the leading coefficient (i.e. the coefficient of the leading monomial) is equal to 1. In the multivariate case, the notion of leading coefficient additionally requires a monomial ordering. For instance, for the order $X_1 > X_2$, the leading monomial of the polynomial $2X_1X_2 + X_1^2$ is X_1^2 and therefore the leading coefficient is 1, whereas the leading monomial in the reverse lexicographic ordering $X_2 > X_1$ is X_1X_2 and the leading coefficient is 2.

Lemma 2. *Given a polynomial $q \in \mathbb{C}[X_1, \ldots, X_n]$, let $p \in \mathbb{C}[X_1, \ldots, X_n]$ be the monic polynomial $\frac{q}{\mathsf{LC}(q)}$, where $\mathsf{LC}(q)$ is the* leading coefficient *of q with respect to some fixed monomial ordering. There exists a non-zero complex number z such that $(zq)^* \in \mathbb{R}[X_1, \ldots, X_n]$ if and only if $p^* \in \mathbb{R}[X_1, \ldots, X_n]$.*

Proof. Suppose there exists such a non-zero complex number z such that $(zq)^* \in \mathbb{R}[X_1, \ldots, X_n]$. Since $z\mathsf{LC}(q) = \mathsf{LC}(zq)$ we have that $\frac{zq}{\mathsf{LC}(zq)} = \frac{zq}{z\mathsf{LC}(q)} = \frac{q}{\mathsf{LC}(q)} = p$, therefore $\frac{1}{\mathsf{LC}(zq)}(zq) = p$ and $\frac{1}{\mathsf{LC}(zq)}(zq)^* = p^*$. Since $\mathsf{LC}(zq) \in \mathbb{R}$, we have $p^* \in \mathbb{R}[X_1, \ldots, X_n]$. Conversely, if $p^* \in \mathbb{R}[X_1, \ldots, X_n]$, take $z = \frac{1}{\mathsf{LC}(q)}$ so that $(zq)^* = p^*$. □

We now describe a procedure for generating decoupling abstract basis polynomials. Suppose we are given all the independent Darboux polynomials in $\mathbb{C}[X_1, \ldots, X_n]$ for the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ up to some degree $d > 0$. By Prop. 2, if there exists a polynomial $p \in \mathbb{R}[X_1, \ldots, X_n]$ of degree $d' \leq d$ such that $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$ is non-constant, then there necessarily exists a Darboux polynomial q of degree d' such that q^* is a polynomial over the reals, i.e. $q^* \in \mathbb{R}[X_1, \ldots, X_n]$. This fact suggests a simple search method. Below we describe the three main steps in the procedure.

1. For a fixed positive integer d, automatically generate all monic Darboux polynomials for the system up to degree d with coefficients in $\mathbb{C}$.
2. For each generated Darboux polynomial q check if $q^* \in \mathbb{R}[X_1, \ldots, X_n]$ and if so, store q^* as a candidate in a list L.
3. For all polynomials q^* in L, run $\textsc{Rewrite}(\mathfrak{L}_f(q^*), q^*)$. If q^* is a decoupling abstract basis polynomial, the rewriting procedure will return $G \in \mathbb{R}[X]$ s.t. $\mathfrak{L}_f(q^*) = G(q^*)$.

Example 4. Consider the following system

$$\dot{x}_1 = \frac{1}{3}(1 + x_1 - 2x_2 + 2(1 + (-1 + x_1 + x_2)^2))$$
$$\dot{x}_2 = \frac{1}{3}(-x_1 + 2x_2 + (-1 + x_1 + x_2)^2)$$

The automatic generation procedure for Darboux polynomials over $\mathbb{C}$ up to degree 1 gives us $(q_1, q_2, q_3) = (1+x_1-2x_2, (-1+\imath)+x_1+x_2, (-1-\imath)+x_1+x_2)$. In this case, q_1^*,q_2^* and q_3^* are all candidates for the short list L. Since $q_2^* = q_3^*$, $L = \{x_1 - 2x_2, x_1 + x_2\}$. Running $\textsc{Rewrite}(\mathfrak{L}_f(q_1^*), q_1^*)$ and $\textsc{Rewrite}(\mathfrak{L}_f(q_2^*), q_2^*)$ returns $2 - 2X + X^2$ and $1 + X$, respectively. Thus, letting $(\alpha_1, \alpha_2) = (q_1^*, q_2^*)$, we obtain the decoupled abstraction:

$$\dot{\alpha}_1 = 2 - 2\alpha_1 + \alpha_1^2,$$
$$\dot{\alpha}_2 = 1 + \alpha_2.$$

In general, a Darboux polynomial q, with $q^* \in \mathbb{R}[X_1, \ldots, X_n]$, is not necessarily a decoupling abstract basis polynomial. For instance, in the system $\dot{x}_1 = x_1x_2, \dot{x}_2 = x_2$, one has x_1 as a Darboux polynomial; however x_1 is not a decoupling abstract basis polynomial because $\mathfrak{L}_f(x_1) = x_1x_2$ cannot be rewritten as polynomial in x_1 only. The checking procedure $\textsc{Rewrite}(\mathfrak{L}_f(x_1), x_1)$ will thus fail to produce a solution.

It is natural to ask under what extra conditions is a Darboux polynomial q satisfying $q^* \in \mathbb{R}[X_1, \ldots, X_n]$ also a decoupling abstract basis polynomial. The following theorem explores this connection.

Theorem 1. *Given a system of polynomial ODEs $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, there exists a polynomial $p \in \mathbb{R}[X_1, \ldots, X_n]$ such that $\mathfrak{L}_f(p) = G(p)$, where $G \in \mathbb{R}[X]$ is of degree $d > 0$, if and only if the system has d Darboux polynomials $q_1, \ldots, q_d \in \mathbb{C}[X_1, \ldots, X_n]$ satisfying:*

(i) $q_1^* = q_2^* = \cdots = q_d^* \in \mathbb{R}[X_1, \ldots, X_n]$,
(ii) $\mathfrak{L}_f(q_1) = \mathfrak{L}_f(q_2) = \cdots = \mathfrak{L}_f(q_d) = rq_1q_2 \cdots q_d$, $r \in \mathbb{R}$,
(iii) *for all $i = 1, \ldots, d$, either $q_i^* - q_i \in \mathbb{R}$ or there exists $j \neq i$, $j = 1, \ldots, d$, such that $q_i = \bar{q}_j$.*

Proof. Suppose there exists a $p \in \mathbb{R}[X_1, \ldots, X_n]$ such that $\mathfrak{L}_f(p) = G(p)$. When $G \in \mathbb{R}[X]$ is a non-constant polynomial of degree d, it can be factorized as $r(X - c_1) \cdots (X - c_d)$, where $r \in \mathbb{R}$ and the roots c_i are either real numbers, or complex numbers that come in conjugate pairs, i.e. if $c_i \in \mathbb{C}$ is a root of G, then its complex conjugate $\bar{c}_i$ is also a root. In the proof of Prop. 2 we have seen that for any such factor $(X - c_i)$ the polynomial $q_i = p - c_i$ is a Darboux polynomial for the system such that $\mathfrak{L}_f(q_i) = G(p)$. The properties (i), (ii) and (iii) follow immediately.

Conversely, let us assume that there are d Darboux polynomials $q_1, q_2, \ldots, q_d$ satisfying properties (i), (ii) and (iii). Then for any $r \in \mathbb{R}$ we have

$$rq_1q_2 \cdots q_d = r(q_1^* - c_1)(q_2^* - c_2) \cdots (q_d^* - c_d),$$

where each $c_i = q_i^* - q_i$ is, by definition, a constant. By property (i) we have $q_1^* = q_2^* = \cdots = q_d^* \in \mathbb{R}[X_1, \ldots, X_n]$, so let us take $p = q_1^* = q_2^* = \cdots = q_d^*$ to obtain $r(q_1^* - c_1)(q_2^* - c_2) \cdots (q_d^* - c_d) = r(p - c_1)(p - c_2) \cdots (p - c_d)$. One can now write this as $r(p - c_1)(p - c_2) \cdots (p - c_d) = G(p)$, where $G \in \mathbb{R}[X]$ has degree d. The coefficients of G are real because by (iii) the roots c_i come in complex conjugate pairs. Since $q_i = q_i^* - (q_i^* - q_i) = p - c_i$, we have $\mathfrak{L}_f(q_i) = \mathfrak{L}_f(p - c_i) = \mathfrak{L}_f(p) - \mathfrak{L}_f(c_i) = \mathfrak{L}_f(p)$ and by (ii) $\mathfrak{L}_f(p) = r(p - c_1)(p - c_2) \cdots (p - c_d) = G(p)$. □

Notice that REWRITE does not require all of the d Darboux polynomials in order to construct G. If a family of Darboux polynomials $\{q_1, \ldots, q_d\}$ as stated in Theorem 1 exists, it suffices to supply only one element, say q_1^*, to REWRITE, which will then find a rewriting of $\mathfrak{L}_f(q_1^*)$ as $G(q_1^*)$, with $G \in \mathbb{R}[X]$. If however, the algorithm fails, then the polynomial supplied was not obtained from such a family of Darboux polynomials and therefore cannot be used to obtain a rewriting of its derivative in terms of itself.

Theorem 1 exposes the structure inherent in systems for which one can find decoupled simulating abstractions. The requirements (i)–(iii) are indeed quite strong. Observe that when G is a linear polynomial with a real coefficient λ, i.e. is of the form $G(X) = \lambda X$ and therefore necessarily has one real root, Theorem 1 reduces to the conditions for constant-scale consecution [23].

Remark 4. Theorem 1 relies on generating Darboux polynomials in order to compute a decoupling abstraction of a given system of polynomial ODEs. Nevertheless, polynomials having *constant* Lie derivatives (that is, those p s.t. $\mathfrak{L}_f(p) = G(p)$ where G has degree zero) can also be used for decoupling abstractions, but are not covered by Theorem 1, which requires the degree of G to be positive. The special case when G has degree zero is also related to Darboux polynomials as follows: (i) when G is the zero polynomial, then the system has a first integral which is a special Darboux polynomial as discussed in Section 2, (ii) when G is a non-zero constant, then the augmented system $(\dot{\boldsymbol{x}}, \dot{t}) = (f(\boldsymbol{x}), 1)$ obtained by appending the time derivative to the original system has a polynomial first integral. More precisely, when $p \in \mathbb{R}[X_1, \ldots, X_n]$ and the $\mathfrak{L}_f(p)$ is a real constant, say r, then in the augmented system $\mathfrak{L}_{(f,1)}(p - rt) = \mathfrak{L}_{(f,1)}(p) - r = \mathfrak{L}_f(p) - r = 0$ and $p - rt$ is thus a polynomial first integral of the augmented system. One may thus handle this case by computing first integrals (e.g. using the approach described in [15]) before searching for more sophisticated decoupling polynomials where G has a positive degree.

4 Outlook

Verification problems for systems of ODEs can be soundly translated to verification problems for their simulating abstractions. Below we sketch the case of a standard safety verification problem (S_x, f, F_x), where one wishes to prove that a given property, encoded as the region $F_x \subset \mathbb{R}^n$, is always satisfied if the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ is initialised in $\boldsymbol{x}_0 \in S_x \subset \mathbb{R}^n$. If a decoupling abstraction $\dot{\boldsymbol{\alpha}} = G(\boldsymbol{\alpha})$

exists, one can attempt to solve the simpler *abstract* safety verification problem (S_y, G, F_y) where $(y_1, \ldots, y_m) = (\alpha_1(\boldsymbol{x}), \ldots, \alpha_m(\boldsymbol{x}))$, denoted henceforth by $\boldsymbol{y} = \boldsymbol{\alpha}(\boldsymbol{x})$, i.e. $\dot{\boldsymbol{y}} = G(\boldsymbol{y})$ is a decoupled simulating abstraction. The initial set in the new abstract coordinates, $S_y \subset \mathbb{R}^m$ (resp. F_y), is computed as a projection of the semialgebraic set $S_x \wedge \boldsymbol{y} = \boldsymbol{\alpha}(\boldsymbol{x})$, which is a subset of $\mathbb{R}^{n+m}$ (resp. $F_x \wedge \boldsymbol{y} = \boldsymbol{\alpha}(\boldsymbol{x})$), onto $\mathbb{R}^m$. Such a projection can in principle be obtained by eliminating the existential quantifiers in the following sentence

$$\exists\, (x_1, \ldots, x_n) \in \mathbb{R}^n . S_x \ \wedge\ y_1 = \alpha_1(x_1, \ldots, x_n) \ \wedge\ \cdots \ \wedge\ y_m = \alpha_m(x_1, \ldots, x_n)\,.$$

The soundness of such an abstraction relies essentially on two facts: (i) the sets S_y and F_y are the exact images through $\boldsymbol{\alpha}$ of the sets S_x and F_x respectively (although using over-approximations of these sets is also sound) and (ii) the invariant regions of the decoupled abstract system, when expressed in terms of the old coordinates, define invariant regions of the original system (i.e. the abstraction is indeed *sound* [22, Theorem 2.2]). This means that if the safety problem holds true in the decoupled abstraction it also holds true in the original concrete system. If not, however, the abstraction may be too coarse.

Interesting directions for refining the abstraction include searching for more general simulating abstractions that are not necessarily completely decoupling. For instance, it is conceivable that a simulating abstraction may possess independent sub-systems that are of the form

$$\begin{aligned} \dot{\alpha}_i &= G_i(\alpha_i, \alpha_j), \\ \dot{\alpha}_j &= G_j(\alpha_i, \alpha_j), \end{aligned}$$

where $G_i, G_j \in \mathbb{R}[X_1, X_2]$ and $\alpha_i, \alpha_j \in \mathbb{R}[X_1, \ldots, X_n]$ are the abstract basis functions. This idea is similar to the so-called algebraizing transformations, briefly discussed in [22, Definition 2.4]. The analysis of 2-dimensional (i.e. planar) polynomial ODEs is however vastly more difficult than the 1-dimensional case. Indeed, qualitative analysis of planar polynomial flows is an active area of mathematical research (e.g. see [6,5]). However, one hope is this greater generality would make simulating abstractions of this form more "common" in systems that one might encounter in applications.

Decoupling can help overcome some of the scalability issues in existing verification methodologies. For instance, in reachability analysis, *relational abstraction* [24] seeks to abstract the flow of a differential equation by an over-approximation of the reachability relation on the states of the system. Mathematically, a (timeless) relational abstraction of an autonomous system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ is a relation $R \subseteq \mathbb{R}^n \times \mathbb{R}^n$ such that $(\boldsymbol{x}, \boldsymbol{y}) \in R$ if $\boldsymbol{y}$ is reachable from $\boldsymbol{x}$ in finite time by following the flow of the system [24, Definition 4], i.e. if $\exists\, t \geq 0.\ \varphi_t(\boldsymbol{x}) = \boldsymbol{y}$. Computing timeless relational abstractions for non-linear systems is difficult because it reduces to searching for positive invariants in the extended system of ODEs $\dot{\boldsymbol{y}} = f(\boldsymbol{y}), \dot{\boldsymbol{x}} = \boldsymbol{0}$ with dimension $2n$, i.e. with twice the number of state variables [24, Definition 5, Lemma 1]. When the system is uncoupled, one can instead work with n extended systems $\dot{y}_i = f_i(y_i), \dot{x}_i = 0$, $i = 1, \ldots, n$, each of dimension 2.

5 Related Work

Our work is closest in spirit to that of Sankaranarayanan [22], which studied simulating abstractions resulting from linearizing change of basis transformations. Our approach instead focused on simulating abstractions obtained via decoupling change of basis transformations.

Change of basis transformations are a standard technique for decoupling linear homogeneous systems of ODEs with constant coefficients, i.e. systems of the form $\dot{\boldsymbol{x}} = A\boldsymbol{x}$, where A is an $n \times n$ real matrix. A common technique applies when the matrix A has n real distinct eigenvalues and produces a decoupled linear homogeneous system $\dot{\boldsymbol{\alpha}} = B\boldsymbol{\alpha}$ of the *same dimension*, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ is made up of *linear functions* $\alpha_i : \mathbb{R}^n \to \mathbb{R}$ in the state variables $x_1, \ldots, x_n$ (see e.g. [21, §28.2, §28.3]); in particular, such a decoupling is always possible when A is a real symmetric matrix. In our work, we consider more general polynomial systems of ODEs and a more general class of polynomials to act as the new basis; additionally, we do not require the dimension of the resulting decoupled system to match that of the original system of coupled ODEs. In short, our focus is not placed on solving the system, but rather on automatically discovering simulating abstractions that are more amenable to analysis.

Girard and Pappas explored *approximate bisimulation* of continuous systems in [9], and Pappas earlier developed (exact) bisimulations between continuous linear systems [18]. However, these works employ a different notion of simulation and do not seek to make the structure of the simulation easier to analyze in the way that we do with decoupling, and are in practice limited to linear ODEs due to reliance on solving linear matrix inequalities (LMIs). Han and Krogh have also explored sound order reduction techniques for verification with reachability analysis, but their approach is also limited to linear ODEs [12]. In contrast to all these existing works that employ different techniques as well as different formal development, our decoupled simulating abstractions are applicable to non-linear polynomial ODEs, and as such, are developed using significantly different methods.

6 Conclusion

In this paper we explored a technique for constructing decoupling simulating abstractions of non-linear polynomial ODEs, which can be more easily analyzed because their 1-dimensional sub-systems may be treated independently. We employed the theory of Darboux polynomials to give a sufficient criterion for non-existence of decoupled simulating abstractions (up to a some maximum degree of the abstract basis polynomials; see Prop. 2). Lastly, we described how automatically generated Darboux polynomials (up to some given polynomial degree) can be used to construct abstract basis polynomials that can yield decoupling simulating abstractions. The abstractions developed in this paper are in essence a form of model transformation, which can be integrated in source transformation and translation tools such as HyST [2]; we leave this for future work.

Acknowledgements The authors would like to thank the anonymous reviewers for their careful reading and judicious critique and extend their thanks to Dr. André Platzer at Carnegie Mellon University for his technical questions and helpful insights into differential ghosts.

Bibliography

1. Abrial, J., Su, W., Zhu, H.: Formalizing hybrid systems with event-b. In: ASM. LNCS, vol. 7316, pp. 178–193. Springer (2012)
2. Bak, S., Bogomolov, S., Johnson, T.T.: HYST: a source transformation and translation tool for hybrid automaton models. In: HSCC. pp. 128–133. ACM (2015)
3. Berz, M., Makino, K.: Verified integration of odes and flows using differential algebraic methods on high-order taylor models. Reliable Computing 4(4), 361–369 (1998)
4. Collins, G.E.: Hauptvortrag: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Automata Theory and Formal Languages, 2nd GI Conference, Kaiserslautern, May 20-23, 1975. LNCS, vol. 33, pp. 134–183. Springer (1975)
5. Conti, R., Galeotti, M.: Dynamical Systems, chap. Totally Bounded Cubic Systems in $\mathbb{R}^2$, pp. 103–171. Springer (2003)
6. Dumortier, F., Llibre, J., Artés, J.C.: Qualitative Theory of Planar Differential Systems. Springer (2006)
7. Ghorbal, K., Platzer, A.: Characterizing algebraic invariants by differential radical invariants. In: TACAS. LNCS, vol. 8413, pp. 279–294. Springer (2014)
8. Ginoux, J.M.: Differential Geometry Applied to Dynamical Systems, World Scientific Series on Nonlinear Science, vol. 66. World Scientific (2009)
9. Girard, A., Pappas, G.J.: Approximate bisimulation: A bridge between computer science and control theory. Eur. J. Control 17(5-6), 568–578 (2011)
10. Goriely, A.: Integrability and Nonintegrability of Dynamical Systems. Advanced series in nonlinear dynamics, World Scientific (2001)
11. Hale, J.K., LaSalle, J.P.: Differential equations: Linearity vs. nonlinearity. SIAM Review 5(3), 249–272 (Jul 1963)
12. Han, Z., Krogh, B.: Reachability analysis of hybrid control systems using reduced-order models. In: American Control Conference, 2004. Proceedings of the 2004. vol. 2, pp. 1183–1189 vol.2 (June 2004)
13. Jeannin, J., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., Platzer, A.: A formally verified hybrid system for the next-generation airborne collision avoidance system. In: TACAS. LNCS, vol. 9035, pp. 21–36. Springer (2015)
14. Johnson, T.T., Green, J., Mitra, S., Dudley, R., Erwin, R.S.: Satellite rendezvous and conjunction avoidance: Case studies in verification of nonlinear hybrid systems. In: Giannakopoulou, D., Méry, D. (eds.) FM 2012: Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings. LNCS, vol. 7436, pp. 252–266. Springer (2012)
15. Matringe, N., Moura, A.V., Rebiha, R.: Generating invariants for non-linear hybrid systems by linear algebraic methods. In: SAS. LNCS, vol. 6337, pp. 373–389. Springer (2010)
16. Nedialkov, N.S.: Interval Tools for ODEs and DAEs. In: 12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN). pp. 4–4 (Sept 2006)

17. Neher, M., Jackson, K.R., Nedialkov, N.S.: On taylor model based integration of odes. SIAM J. Numerical Analysis 45(1), 236–262 (2007)
18. Pappas, G.J.: Bisimilar linear systems. Automatica 39(12), 2035–2047 (2003)
19. Platzer, A.: A complete uniform substitution calculus for differential dynamic logic. Journal of Automated Reasoning pp. 1–47 (2016)
20. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: FM. LNCS, vol. 5850, pp. 547–562. Springer (2009)
21. Robinson, J.C.: An introduction to ordinary differential equations. Cambridge University Press (2004)
22. Sankaranarayanan, S.: Change-of-bases abstractions for non-linear hybrid systems. Nonlinear Analysis: Hybrid Systems 19, 107 – 133 (2016)
23. Sankaranarayanan, S., Sipma, H.B., Manna, Z.: Constructing invariants for hybrid systems. Formal Methods in System Design 32(1), 25–55 (2008)
24. Sankaranarayanan, S., Tiwari, A.: Relational abstractions for continuous and hybrid systems. In: CAV. LNCS, vol. 6806, pp. 686–702. Springer (2011)
25. Strogatz, S.H.: Nonlinear Dynamics and Chaos. Westview Press (1994)
26. Tarski, A.: A decision method for elementary algebra and geometry. Bulletin of the American Mathematical Society 59 (1951)
27. Teschl, G.: Ordinary Differential Equations and Dynamical Systems, Graduate Studies in Mathematics, vol. 140. American Mathematical Society (2012)
28. Zhao, H., Yang, M., Zhan, N., Gu, B., Zou, L., Chen, Y.: Formal verification of a descent guidance control program of a lunar lander. In: FM. LNCS, vol. 8442, pp. 733–748. Springer (2014)
29. Zou, L., Lv, J., Wang, S., Zhan, N., Tang, T., Yuan, L., Liu, Y.: Verifying chinese train control system under a combined scenario by theorem proving. In: VSTTE. LNCS, vol. 8164, pp. 262–280. Springer (2013)